

Tight Bounds for an Online Bin Packing Problem*

Joan Boyar[†]

Faith Ellen[‡]

April 30, 2014

Abstract

The following online bin packing problem is considered: Items with integer sizes are given and variable sized bins arrive online. A bin must be used if there is still an item remaining which fits in it when the bin arrives. The goal is to minimize the total size of all the bins used. Previously, a lower bound of $\frac{5}{4}$ on the competitive ratio of this problem was achieved using jobs of size S and $2S - 1$ and maximum bin size $4S - 4$. For this case, we obtain matching upper and lower bounds, which vary depending on the ratio of the number of small jobs to the number of large jobs.

1 Introduction

In the classical online bin packing problem, bins of unit size are given, and items of at most unit size arrive online. The goal is to pack the items using as few bins as possible. This is reversed in both Zhang’s Bin Packing problem [10] and the Grid Scheduling problem introduced in [1], where items and their sizes are given at the beginning and variable sized bins arrive in an online manner. In both problems, the items must be packed in the bins, such that the total size of the items packed in a bin is no more than the size of the bin, and the goal is to pack all items, minimizing the total size of bins used.

In Zhang’s Bin Packing problem, bins and items can have any sizes in the range $(0, 1]$, and the largest item is no larger than the smallest bin. All bins which arrive while there are still unpacked items are considered “used”, even if the algorithm, for some reason, puts no items in some bins.

*A preliminary version of this paper, entitled “Bounds for Scheduling Jobs on Grid Processors”, appeared in *Space-Efficient Data Structures, Streams, and Algorithms*, Lecture Notes in Computer Science, volume 8066, 2013, pages 12-26.

[†]Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark, joan@imada.sdu.dk. Supported in part by the Villum Foundation and the Danish Council for Independent Research, Natural Sciences (FNU). Part of this work was carried out while this author was visiting the University of Waterloo and the University of Toronto.

[‡]Department of Computer Science, University of Toronto, 10 King’s College Road, Toronto, Ontario, Canada M5S 3G4, faith@cs.toronto.edu. Supported in part by the VELUX Foundation and the Natural Science and Engineering Research Council of Canada (NSERC). Part of this work was carried out while this author was visiting the University of Southern Denmark.

In contrast, the Grid Scheduling problem, which we consider, does not have the restriction that the largest item is no larger than the smallest bin. However, the algorithm is required to use a bin as long as there is at least one unpacked item which fits in the bin when it arrives. In addition, the bins and items have integer sizes between 1 and some maximum size M . If there is no bound on the maximum bin size, the problem is uninteresting, because the competitive ratio is unbounded: Once enough bins for an optimal packing have arrived, an adversary could send bins of arbitrarily large size, which the algorithm would be forced to use.

The original motivation for the Grid Scheduling problem in [1, 2, 4] was the following: A large problem, (for example, from the well-known bioinformatics program BLAST) is divided up into independent jobs, possibly with varying sizes. These jobs are distributed to heterogeneous processors having idle time (for example, personal computers). These processors become available in an on-line manner, as they become idle. Folding@Home (in which idle processors are used to find new ways to fold proteins) and SETI@home (in which idle processors are used to look for signs of extraterrestrial life) are two examples. Such systems have been called Grids, denoting computation distributed seamlessly over (possibly) large distances, in the same way that electricity is distributed over the electrical grid. It does not refer to computation in which the network topology is a rectangular grid. Furthermore, the terminology Grid scheduling has also been used to denote other problems which are very different from the one considered here.

It is well-known that paging may slow down computation drastically. In fact, parallelizing jobs can result in superlinear speed-up by eliminating unnecessary paging [3]. Paging can be avoided by assigning jobs to a processor whose memory capacity is at least the combined memory requirements of those jobs. In the Grid Scheduling problem, the item sizes are the memory requirements of the jobs and the bin sizes are the memory capacity of the processors. To finish the jobs as quickly as possible, any processor that arrives and can process at least one of the remaining jobs should be assigned to it (rather than waiting for a processor of close to the right size, which might never come). For the Grid Scheduling problem, we show that this condition is equivalent to requiring that any arriving bin should be assigned items until there are no more items that can fit into it.

In [1], Boyar and Favrholdt presented an algorithm to solve this problem with competitive ratio $\frac{13}{7}$. This algorithm also solves Zhang's Bin Packing problem with the same competitive ratio, solving an open question proposed in [10]. Grid Scheduling with Conflicts, the Grid Scheduling problem with the restriction that certain pairs of jobs cannot be executed on the same processor, was later studied by Epstein et al. in [4].

Boyar and Favrholdt [1] also proved that the competitive ratio of any algorithm for the Grid Scheduling problem is at least $\frac{5}{4}$. (Note that, although upper bounds for the Grid Scheduling problem also apply to Zhang's Bin Packing problem, lower bounds do not apply, since the problem is more general.) In that proof, the adversary used only two different item sizes, S and $L = 2S - 1$, and four different bin sizes, S , $2S - 1$, $2S$, and $4S - 4$, for an integer $S > 1$. The adversary gives n items of size L and $2n$ of size S , and initially gives n bins of size $2S$. The algorithm then fills some number of these bins with items of size L and the remaining with (at most) two items of size S each. Depending on the algorithm's behavior, the adversary can use either all its items of size S or all of its items of size L in this set of bins. Then, it can arrange that all of its remaining items are placed in bins where they fit exactly, with no empty space, while the algorithm will have empty space equal to almost twice the item size for at least half of its original items of size S or original items of size L . This gives the $\frac{5}{4}$ lower bound. It is crucial that L is almost twice the size of S for

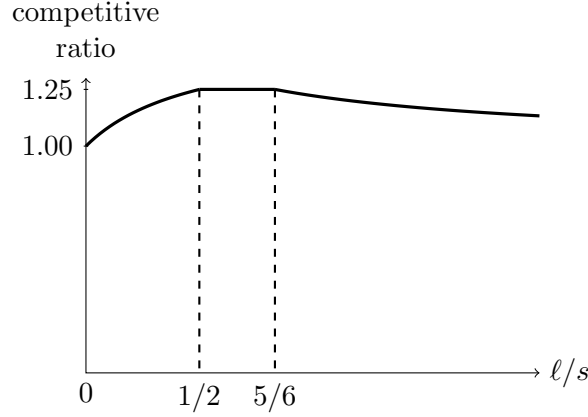


Figure 1: The competitive ratio of the Restricted Grid Scheduling problem as a function of ℓ/s

two reasons. First, the original bins need to be approximately the same size as the items of size L but still allow two items of size S to fit there. Second, in the case where the adversary used items of size S in the first set of bins, it must give bins of size (at least) L for its remaining items, and the algorithm must place only one item of size S in each of these bin leaving empty space almost S .

In this paper, we consider the Restricted Grid Scheduling problem, a version of the Grid Scheduling problem where the input contains exactly s items of size $S > 1$ and ℓ items of size $L = 2S - 1$ and the maximum bin size is $M = 4S - 3$. This includes the scenarios used in the $\frac{5}{4}$ lower bound. Two natural questions arise for this problem:

1. Is there an algorithm matching the $\frac{5}{4}$ lower bound on the competitive ratio or can this lower bound be improved?
2. What is the competitive ratio for the problem when there are initially s items of size S and ℓ items of size L , when the ratio of s to ℓ is arbitrary, rather than fixed to 2?

We obtain matching upper and lower bounds on the competitive ratio of the Restricted Grid Scheduling problem. Our bound is plotted in Figure 1.

Theorem 1 *For unbounded S , the competitive ratio of the Restricted Grid Scheduling problem is*

$$\begin{cases} 1 + \frac{1}{2+s/\ell} & \text{if } \ell \leq s/2, \\ 1 + \frac{1}{4} & \text{if } s/2 < \ell \leq 5s/6, \text{ and} \\ 1 + \frac{2}{3+6\ell/s} & \text{if } 5s/6 < \ell. \end{cases}$$

In many applications of bin packing, there are only a small number of different item sizes. A number of papers have considered the problem of packing a sequence of items of two different sizes in bins of size 1 in an online matter. In particular, there is a lower bound of $4/3$ on the competitive

ratio [8, 6] and a matching upper bound [6]. When both item sizes are bounded above by $1/k$, the competitive ratio can be improved to $\frac{(k+1)^2}{k^2+k+1}$ [5].

We begin with some preliminaries, including a formal definition of the Grid Scheduling problem and some properties of optimal packings. In Section 3, we prove the lower bound on the competitive ratio of the Restricted Grid Scheduling problem (and, hence, of the Grid Scheduling problem). Then we provide a matching upper bound, which proves that this lower bound for the Restricted Grid Scheduling problem cannot be improved. It also shows that any better lower bound for the Grid Scheduling problem must use more or different item sizes or a larger maximum bin size. In Section 4, we present properties of optimal packings for the Restricted Grid Scheduling problem. Some of these provide motivation for the design of our algorithm, while others are important for the analysis. We show why a few simple algorithms are not optimal in Section 5. Our algorithm, 2-Phase-Packer, appears in Section 6, together with the analysis. We conclude with some open questions.

2 Preliminaries

The competitive ratio [9, 7] of an on-line algorithm is the worst-case ratio of the on-line performance to the optimal off-line performance, up to an additive constant. More precisely, for a set I of items (or, equivalently, a multi-set of item sizes), a sequence σ of bins, and an algorithm \mathbb{A} for the Grid Scheduling problem, let $\mathbb{A}(I, \sigma)$ denote the total size of all the bins used by \mathbb{A} when packing I in the sequence σ of bins. Then, the *competitive ratio* $\text{CR}_{\mathbb{A}}$ of \mathbb{A} is

$$\text{CR}_{\mathbb{A}} = \inf \{c \mid \exists b, \forall I, \forall \sigma, \mathbb{A}(I, \sigma) \leq c \cdot \text{OPT}(I, \sigma) + b\},$$

where OPT denotes an optimal off-line algorithm. For specific choices of families of sets I_n and sequences σ_n , the performance ratios, $\frac{\mathbb{A}(I_n, \sigma_n)}{\text{OPT}(I_n, \sigma_n)}$, can be used to prove a lower bound on the competitive ratio of \mathbb{A} .

Given a set of items, each with a positive integer size, and a sequence of bins, each with a positive integer size at most M , the goal of the Grid Scheduling problem is to pack the items in the bins so that the sum of the sizes of the items packed in each bin is at most the size of the bin and the sum of the sizes of bins used is minimized. The bins in the sequence arrive one at a time and each must be packed before the next bin arrives, without knowledge of the sizes of any future bins. If a bin is at least as large as the smallest unpacked item, it must be packed with at least one item. There is no charge for a bin that is smaller than this. Note there is no loss of generality in assuming that every item has size at most M , because no item of size greater than M can be packed.

Because all items are required to be packed, it is assumed that enough sufficiently large bins arrive. Thus any algorithm eventually packs all items. For example, it suffices that every sequence ends with enough bins of size M to pack every item one per bin.

Given a set of items and a sequence of bins, a *partial packing* is an assignment of some of the items to bins such that the sum of the sizes of the items assigned to each bin is at most the size of the bin. A partial packing is *valid* if each empty bin is smaller than each unpacked item and smaller than each item packed in a bin that occurs later in the sequence. Thus, each empty bin in a valid partial packing is smaller than all items that were unpacked at the time the bin arrived. A *packing*

is a partial packing that assigns every item to a bin. In a valid packing, each empty bin is smaller than all items packed in later bins.

Lemma 2 *For any set of items I , any item $i \in I$, and any sequence of bins σ , if p is a valid packing of I into σ , then there is a valid packing of $I - \{i\}$ into σ that uses a subset of the bins used by p .*

Proof: Suppose not. Consider a counterexample where $|I|$ is minimum, consisting of a valid packing p of a set of items I into a sequence of bins $\sigma = \langle b_1, b_2, \dots, b_m \rangle$ and an item $i \in I$.

Since p is valid, if p does not use bin b_1 , then all items in I have size larger than $\text{size}(b_1)$ and so b_1 is used in no packing of I . Hence, removing b_1 from σ still results in a counterexample. Therefore, without loss of generality, we may assume that p uses the first bin of σ .

Let J be the set of items that p packs into b_1 . Let $\sigma' = \langle b_2, \dots, b_m \rangle$ be obtained from σ by removing its first bin. Let q be the packing of $I - J$ that packs every item into the same bin as p does. Then q is a valid packing of $I - J$ into σ' .

If $i \notin J$, then, since $|I - J| < |I|$, there is a valid packing q' of $I - J - \{i\}$ into σ' that uses a subset of the bins used by q . Let q'' be the packing of $I - \{i\}$ into σ that packs the items in J into bin b_1 and packs the items of $I - J - \{i\}$ into the same bins as q' does. Then q'' is valid and uses a subset of the bins used by p , contradicting the choice of p . Therefore $i \in J$.

Consider the packing p' of $I - \{i\}$ into σ that packs every item into the same bins as p does. If $J \neq \{i\}$ or all items in $I - \{i\}$ are larger than b_1 , then p' is valid and uses a subset of the bins used by p , a contradiction. Therefore $J = \{i\}$ and there exists an item $j \in I - \{i\}$ whose size is at most $\text{size}(b_1)$.

Note that p' is a valid packing of $I - \{i\}$ into σ' . Since $I - \{i\}$ is smaller than I , there is a valid packing p'' of $(I - \{i\}) - \{j\} = I - \{i, j\}$ into σ' that uses a subset of the bins used by p' . Let p''' be the packing of $I - \{i\}$ into σ that packs item j into bin b_1 and packs the items of $I - \{i, j\}$ into the same bins as p'' does. Then p''' is valid and uses a subset of the bins used by p , a contradiction. ■

A bin in a partial packing is *wasteful* if its empty space is at least as large as some unpacked item or some item packed in a bin that occurs later in the sequence. In other words, when this bin was packed, there was an extra item that could have been added to it, but was not. A partial packing is *thrifty* if it contains no wasteful bins. Since a partial packing is valid if and only if it has no wasteful empty bin, every thrifty partial packing is valid.

The *cost* of a packing is the sum of the sizes of the bins it uses (i.e. in which it places at least one item). A packing is *optimal* (among valid packings) if it is valid and its cost is at least as small as any other valid packing for the same set of items and sequence of bins. Note that there may be invalid packings of smaller cost. For example, with two items of size S , an optimal packing for the sequence of bins $\langle S, 2S \rangle$ puts one item in each bin.

The following result says that we may restrict attention to thrifty packings.

Lemma 3 *For any valid packing p of a set of items into a sequence of bins, there is a thrifty packing of those items into that sequence using a subset of the bins used by p .*

Proof: Let p be an valid packing of a set of items I into a sequence of bins σ and let Q be the set of valid packings of I into σ that use a subset of the bins used by p .

To obtain a contradiction, suppose that there is no thrifty packing in Q . Then every packing in Q has at least one wasteful bin. Consider the packings in Q whose first wasteful bin occurs latest in σ . Let b be this bin. Among all packings in Q whose first wasteful bin is b , let q be one which has the least amount, e , of empty space in bin b .

Since b is a wasteful bin of q , there is some item, i , of size at most e that q packs in a bin that occurs after b in σ . Let σ' be the sequence of bins in σ following b , let I' be the set of items q packs into bins that follow b , and let p' be the valid packing of I' into σ' that packs each item into the same bin as q . By Lemma 2, there is a valid packing p'' of $I' - \{i\}$ into σ' that uses a subset of the bins used by p' .

Let q' be the packing of I into σ that packs each item in $I - I'$ in the same bin that q does, packs item i in bin b , and packs each item in $I' - \{i\}$ in the same bin that p'' does. Then q' is a valid packing that uses a subset of the bins used by q , which is a subset of the bins used by p . Thus $q' \in Q$. However, no bin before b is wasteful in q' and bin b has less empty space in q' than it does in q . Thus q' contradicts the choice of q . ■

Corollary 4 *For any optimal packing of a set of items into a sequence of bins, there is an optimal thrifty packing of those items into that sequence using the same set of bins.*

Proof: Let p be an optimal packing of a set of items I into a sequence of bins σ . Since p is a valid packing, Lemma 3 implies that there is a thrifty packing q of I into σ using a subset of the bins used by p . Thus $\text{cost}(q) \leq \text{cost}(p)$. But p is optimal, so $\text{cost}(q) = \text{cost}(p)$. Hence q is optimal and uses the same set of bins that p uses. ■

The Restricted Grid Scheduling problem is the restriction of the Grid Scheduling problem in which all items have size S or $L = 2S - 1$, where $M = 4S - 3$ and $S > 1$. There is no loss of generality in assuming that every bin has size at least S , because no bin of size less than S can be used to pack an item.

3 Lower Bounds

Theorem 5 *For unbounded S , any algorithm for the Restricted Grid Scheduling problem has competitive ratio at least*

$$\begin{cases} 1 + \frac{1}{2+s/\ell} & \text{if } \ell \leq s/2, \\ 1 + \frac{1}{4} & \text{if } s/2 < \ell \leq 5s/6, \text{ and} \\ 1 + \frac{2}{3+6\ell/s} & \text{if } 5s/6 < \ell. \end{cases}$$

Proof: Consider an algorithm for the Restricted Grid Scheduling problem and an instance in which there are s items of size S , ℓ items of size $L = 2S - 1$, and maximum bin size $M = 2L - 1$. By

Lemma 3, we may assume the algorithm is thrifty. We start with the case when $\ell \leq s/2$ and then handle the case when $\ell > s/2$. In both cases, we consider two subcases, depending on how the algorithm packs the first batch of bins.

Case I: $\ell \leq s/2$.

The adversary begins by giving ℓ bins of size $2S$. In each of these bins, the algorithm must pack either two items of size S or one item of size L . Let $0 \leq k \leq \ell$ be the number of these bins in which the algorithm packs two items of size S . Then the algorithm has $s - 2k$ items of size S and k items of size L left to pack.

Case I.1: $k \leq \ell/2$.

Next, the adversary gives $s - 2\ell$ bins of size S , followed by 2ℓ bins of size L . The algorithm must pack one item of size S in each bin of size S and must use one bin of size L for each of the remaining $s - 2k - (s - 2\ell) = 2(\ell - k)$ items of size S and k items of size L . The total cost incurred by the algorithm is $\ell \cdot 2S + (s - 2\ell) \cdot S + (2\ell - k) \cdot L = s \cdot S + 2\ell \cdot L - k \cdot L \geq s \cdot S + 3\ell \cdot L/2$.

For this sequence, OPT packs two items of size S in each of the ℓ bins of size $2S$, one item of size S in each of the $s - 2\ell$ bins of size S , and one item of size L in each of the next ℓ bins of size L , for total cost $s \cdot S + \ell \cdot L$. Thus, the performance ratio of the algorithm is at least

$$\begin{aligned} \frac{s \cdot S + 3\ell \cdot L/2}{s \cdot S + \ell \cdot L} &= 1 + \frac{\ell \cdot L}{2\ell \cdot L + 2s \cdot S} = 1 + \frac{\ell}{2\ell + s + s/L} \\ &= 1 + \frac{1}{2 + \frac{s}{\ell} + \frac{s/\ell}{2S-1}} \rightarrow 1 + \frac{1}{2 + s/\ell} \text{ as } S \rightarrow \infty. \end{aligned}$$

Case I.2: $k > \ell/2$.

Next, the adversary gives s bins of size S , followed by ℓ bins of size M . The algorithm packs one item of size S in the first $s - 2k \geq 2\ell - 2k \geq 0$ of these bins, using up all its items of size S . It discards the remaining $2k$ bins of size S , because it has no remaining elements that are small enough to fit in them. Then the algorithm packs its remaining k items of size L into k bins of size $M = 4S - 3$. The total cost incurred by the algorithm is

$$\begin{aligned} &\ell \cdot 2S + (s - 2k) \cdot S + k \cdot (4S - 3) \\ &= (2\ell + s) \cdot S + k \cdot (2S - 3) \\ &> (2\ell + s) \cdot S + \ell \cdot (S - 3/2). \end{aligned}$$

For this sequence, OPT packs one item of size L in each of the ℓ bins of size $2S$ and one item of size S in each of the next s bins. The total cost used by OPT is $\ell \cdot 2S + s \cdot S$. Thus, the performance ratio of the algorithm is at least

$$\frac{(3\ell + s) \cdot S - 3\ell/2}{(2\ell + s) \cdot S} = 1 + \frac{1 - 3/(2S)}{2 + s/\ell} \rightarrow 1 + \frac{1}{2 + s/\ell} \text{ as } S \rightarrow \infty.$$

Case II: $\ell > s/2$.

The adversary begins by giving $\lfloor s/2 \rfloor$ bins of size $2S$. In each of these bins, the algorithm must pack either two items of size S or one item of size L . Let $0 \leq k \leq \lfloor s/2 \rfloor$ be the number of these

bins in which the algorithm packs two items of size S . Then the algorithm has $s - 2k$ items of size S and $\ell - \lfloor s/2 \rfloor + k$ items of size L left to pack.

Case II.1: $k \leq \lfloor s/2 \rfloor - s/8 - \ell/4 + 1$ or $k \leq \lfloor s/2 \rfloor - s/3 + 1$.

Next, the adversary next gives $\lceil s/2 \rceil - \lfloor s/2 \rfloor$ bins of size S (i.e. one bin of size S if s is odd and no bins of size S if s is even), $\lfloor s/2 \rfloor - k + \ell - 1$ bins of size L , and 1 bin of size M . Since $(s - 2k) + (\ell - \lfloor s/2 \rfloor + k) = (\lceil s/2 \rceil - \lfloor s/2 \rfloor) + (\lfloor s/2 \rfloor - k + \ell - 1) + 1$, the algorithm packs one of its remaining items in each of these bins, so the total cost it incurs is $\lfloor s/2 \rfloor \cdot 2S + (\lceil s/2 \rceil - \lfloor s/2 \rfloor) \cdot S + (\lfloor s/2 \rfloor - k + \ell - 1) \cdot L + M = s \cdot S + \ell \cdot L + (\lfloor s/2 \rfloor - k + 1) \cdot L - 1$.

If $s \geq 4$, then $\lfloor s/2 \rfloor - k \geq \min\{s/8 + \ell/4, s/3\} - 1 > s/4 - 1 \geq 0$, so there are at least ℓ bins of size L . Because of the additive constant in the definition of the competitive ratio, the case $s \leq 3$ can be ignored. For this sequence, OPT packs two items of size S in each of the $\lfloor s/2 \rfloor$ bins of size $2S$, one item of size S in the bin of size S , if s is odd, and one item of size L in each of the next ℓ bins of size L . Its total cost is $s \cdot S + \ell \cdot L$.

If $k \leq \lfloor s/2 \rfloor - s/8 - \ell/4 + 1$, then $(\lfloor s/2 \rfloor - k + 1) \cdot L \geq (s/8 + \ell/4) \cdot L = S \cdot s/4 - s/8 + L \cdot \ell/4$, so the performance ratio of the algorithm is at least

$$\frac{(s \cdot S + \ell \cdot L) \cdot 5/4 - s/8 - 1}{s \cdot S + \ell \cdot L} = \frac{5}{4} - \frac{s/8 + 1}{s \cdot S + \ell \cdot (2S - 1)} \rightarrow \frac{5}{4} \text{ as } S \rightarrow \infty.$$

Similarly, if $k \leq \lfloor s/2 \rfloor - s/3 + 1$, then $(\lfloor s/2 \rfloor - k + 1) \cdot L \geq (s/3) \cdot L$, so the performance ratio of the algorithm is at least

$$1 + \frac{(s/3) \cdot L}{s \cdot S + \ell \cdot L} = 1 + \frac{2 - 1/(s \cdot S)}{3 + 6\ell/s - 3\ell/(s \cdot S)} \rightarrow 1 + \frac{2}{3 + 6\ell/s} \text{ as } S \rightarrow \infty.$$

Case II.2: $k > \lfloor s/2 \rfloor - s/8 - \ell/4 + 1$ and $k > \lfloor s/2 \rfloor - s/3 + 1 > s/6$.

Next, the adversary gives $\max\{s - 2k, s - \ell + \lfloor s/2 \rfloor\}$ bins of size S , followed by $\min\{2k, \ell - \lfloor s/2 \rfloor\}$ bins of size $L + S$, $\max\{\ell - \lfloor s/2 \rfloor - 2k, 0\}$ bins of size L , and finally k bins of size $M = 4S - 3$. The algorithm packs its $s - 2k$ items of size S into bins of size S . Since $\min\{2k, \ell - \lfloor s/2 \rfloor\} + \max\{\ell - \lfloor s/2 \rfloor - 2k, 0\} + k = \ell - \lfloor s/2 \rfloor + k$, the algorithm packs one item of size L in each bin of size $L + S$, L , and M . The total cost incurred by the algorithm is

$$\begin{aligned} & \lfloor s/2 \rfloor \cdot 2S + (s - 2k) \cdot S + \min\{2k, \ell - \lfloor s/2 \rfloor\} \cdot (L + S) \\ & + \max\{\ell - \lfloor s/2 \rfloor - 2k, 0\} \cdot L + k \cdot M \\ & = (2\lfloor s/2 \rfloor + s) \cdot S + (\ell - \lfloor s/2 \rfloor) \cdot L + k \cdot (2S - 3) + \min\{2k, \ell - \lfloor s/2 \rfloor\} \cdot S. \end{aligned}$$

For this sequence, OPT fills every bin it uses except for the $\lfloor s/2 \rfloor$ bins of size $2S$, in which it puts items of size $L = 2S - 1$. Therefore, the total cost used by OPT is $s \cdot S + \ell \cdot L + \lfloor s/2 \rfloor$.

If $2k \geq \ell - \lfloor s/2 \rfloor$, the performance ratio of the algorithm is at least

$$\begin{aligned} & \frac{(2\lfloor s/2 \rfloor + s) \cdot S + (\ell - \lfloor s/2 \rfloor) \cdot L + k \cdot (2S - 3) + (\ell - \lfloor s/2 \rfloor) \cdot S}{s \cdot S + \ell \cdot L + \lfloor s/2 \rfloor} \\ & > \frac{(\ell + s + \lfloor s/2 \rfloor) \cdot S + (\ell - \lfloor s/2 \rfloor) \cdot L + (\lfloor s/2 \rfloor - s/8 - \ell/4 + 1) \cdot (2S - 3)}{s \cdot S + \ell \cdot L + \lfloor s/2 \rfloor} \\ & = \frac{5}{4} + \frac{(\lfloor s/2 \rfloor - s/2 + 2) \cdot S + \ell + 3s/8 - 13\lfloor s/2 \rfloor/4 - 3}{(s + 2\ell) \cdot S - \ell + \lfloor s/2 \rfloor} \\ & > \frac{5}{4} + \frac{\ell + 3s/8 - 13\lfloor s/2 \rfloor/4 - 3}{(s + 2\ell) \cdot S - \ell + \lfloor s/2 \rfloor} \rightarrow \frac{5}{4} \text{ as } S \rightarrow \infty. \end{aligned}$$

If $2k \leq \ell - \lfloor s/2 \rfloor$, the performance ratio of the algorithm is at least

$$\begin{aligned}
& \frac{(2\lfloor s/2 \rfloor + s) \cdot S + (\ell - \lfloor s/2 \rfloor) \cdot L + k \cdot (4S - 3)}{s \cdot S + \ell \cdot L + \lfloor s/2 \rfloor} \\
&= 1 + \frac{k \cdot (4S - 3)}{(s + 2\ell) \cdot S - \ell + \lfloor s/2 \rfloor} \\
&> 1 + \frac{(s/6) \cdot (4S - 3)}{(s + 2\ell) \cdot S + \lfloor s/2 \rfloor - \ell} \\
&= 1 + \frac{2 - 3/2S}{3 + 6\ell/s + 3(\lfloor s/2 \rfloor - \ell)/sS} \rightarrow 1 + \frac{2}{3 + 6\ell/s} \text{ as } S \rightarrow \infty.
\end{aligned}$$

Note that $\frac{5}{4} \leq 1 + \frac{2}{3 + 6\ell/s}$ if and only if $\ell \leq 5s/6$. Thus, $\frac{5}{4}$ is a lower bound on the competitive ratio when $s/2 < \ell \leq 5s/6$ and $1 + \frac{2}{3 + 6\ell/s}$ is a lower bound on the competitive ratio when $\ell > 5s/6$. ■

4 Properties of Optimal Packings for Restricted Grid Scheduling

We say that a bin used in a packing is *bad* if it contains at least one item of size S , it has empty space at least $L - S$, and it occurs while items of size L remain. Note that a bin containing an item of size L and an item of size S has empty space at most $M - L - S = L - S - 1$, so it is not bad.

Lemma 6 *For any set of items and any sequence of bins, there exists an optimal thrifty packing that contains no bad bin.*

Proof: Fix a sequence of bins σ (each of size at least S) and a set of items I . To obtain a contradiction, suppose that every optimal thrifty packing of I into σ contains a bad bin. Consider an optimal thrifty packing p which has the longest prefix without bad bins. Let b' be the last bin to which p assigns an item of size L . Then the first bad bin b occurs before b' . Since p is thrifty, the empty space in b is less than L .

Suppose p has an empty bin between b and b' . Let b'' be the first empty bin following b . Since p is valid, $\text{size}(b'') < L$ and, when bin b'' arrives, only items of size L remain. Then each nonempty bin after b'' , including b' , contains exactly one item, which is of size L . Consider the packing p' obtained from p by moving one item of size S from bin b to bin b'' and moving the item of size L in bin b' to bin b . Then b' is empty in the packing p' . Since p is valid, p and p' are the same prior to bin b , there are no empty bins between b and b'' in p' , and no item which occurred before b'' in p has been moved after b'' in p' , it follows that p' is valid. But the cost of p' is equal to $\text{cost}(p) - \text{size}(b') + \text{size}(b'') < \text{cost}(p)$, since $\text{size}(b'') < L \leq \text{size}(b')$. This contradicts the optimality of p . Therefore p has no empty bins between b and b' .

Let p' be the packing obtained from p by switching an item of size S in b with the item of size L in b' . Note that p' is optimal, since p is. Since b contains an item of size L , it is not bad in p' . Furthermore, bin b is not wasteful, since its empty space, which was less than L in p , is less than $L + S - L = S$ in p' . None of the bins in p are wasteful, since p is thrifty, so none of the bins in p' prior to b are wasteful.

Let J be the set of items that p' packs into bins up to and including b , let σ' be the suffix of σ following bin b , and let q be the packing of $I - J$ into σ' that packs every item in $I - J$ into the same bin that p' does. Then q is an optimal packing of I into σ . By Corollary 4, there is an optimal thrifty packing q'' of $I - J$ into σ' that uses the same bins as q . Let p'' be the packing of I into σ that packs each item in J into the same bin that p' does and packs each item in $I - J$ into the same bin that q'' does. Then p'' is an optimal thrifty packing. But p'' has a longer prefix without bad bins than p does. Thus, it contradicts the choice of p . ■

This motivates the following definition.

Definition 7 *A partial packing is reasonable if every bin b contains*

- *one item of size S , if $\text{size}(b) \in [S, L - 1]$,*
- *one item of size L , if $\text{size}(b) = L$,*
- *two items of size S or one item of size L , if $\text{size}(b) \in [L + 1, L + S - 1]$,*
- *one item of size S and one item of size L , if $\text{size}(b) = L + S$, and*
- *three items of size S or one item of size S and one item of size L , if $\text{size}(b) \in [L + S + 1, 2L - 1]$.*

Note that a reasonable partial packing contains no wasteful, bad, or empty bins.

For any packing, consider the first bin after which there are no items of size L remaining or at most 2 items of size S remaining. This bin is called the *key bin* of the packing. The partial packing that assigns the same items into each bin up to and including its key bin and assigns no items to any subsequent bin is called the *front* of the packing. We say that a packing is *reasonable* if it is thrifty and its front is reasonable.

Corollary 8 *For any set of items and any sequence of bins, there exists an optimal packing that is reasonable.*

From now on, we will restrict attention to reasonable packings.

Given a set of items and a sequence of bins, two reasonable partial packings can differ as to whether they use one item of size L or two items of size S in certain bins. Therefore, the numbers of items of size S and items of size L they do not assign may differ. However, if both have at least one item of size S and at least one item of size L unassigned, the set of bins they have used is the same and there is a simple invariant relating the numbers of unassigned items of size S and unassigned items of size L they have.

Lemma 9 *Consider two reasonable partial packings of a set of items into a sequence of bins. Suppose that before bin b , each packing has at least one item of size S and at least one item of size L available. Then immediately after bin b has been packed, the sum of the number of items of size S available plus twice the number of items of size L available is the same for both packings.*

Proof: Consider any bin b in the sequence σ and suppose that immediately before bin b is packed, each packing has items of both size S and L available and the sum of the number of items of size S available plus twice the number of items of size L available is the same for both packings. Note that this is true initially, since all items are still available for both packings. Since both packings are reasonable, either bin b is filled the same way in both packings or in one of these packings, bin b contains two more items of size S and one less item of size L than the other. Thus the claim remains true immediately after bin b is packed. ■

For any sequence of bins σ and any nonnegative integers s and ℓ , let $OPT(\sigma, s, \ell)$ denote the cost of an optimal packing of s items of size S and ℓ items of size $L = 2S - 1$ using σ . This must be at least the sum of the sizes of all the items.

Proposition 10 *For all sequences of bins σ and all integers $s, \ell \geq 0$, $OPT(\sigma, s, \ell) \geq sS + \ell L$.*

Given any optimal packing for a set of items, a packing for a subset of these items can be obtained by removing the additional items from bins, starting from the end.

Proposition 11 *For all sequences of bins σ and all integers $0 \leq s' \leq s$ and $0 \leq \ell' \leq \ell$, $OPT(\sigma, s', \ell') \leq OPT(\sigma, s, \ell)$.*

For any sequence of bins σ and any nonnegative integers s and ℓ , let $R(\sigma, s, \ell)$ denote the maximum cost of any reasonable packing of s items of size S and ℓ items of size $L = 2S - 1$ using σ .

When all items have the same size, all thrifty algorithms, including OPT, behave exactly the same.

Proposition 12 *For all sequences of bins σ and all integers $s', \ell' \geq 0$, $R(\sigma, s', 0) = OPT(\sigma, s', 0)$ and $R(\sigma, 0, \ell') = OPT(\sigma, 0, \ell')$.*

Thus, if R and OPT both run out of items of size L at the same time or they both run out of items of size S at the same time, while having items of the other size remaining, then they will use the same set of bins and, hence, have the same cost. The following four lemmas describe the relationship between the costs incurred by R and OPT when one of them has run out of one size of items.

Lemma 13 *For all sequences of bins σ and all integers $s', \ell' \geq 0$, $R(\sigma, s', \ell') \leq OPT(\sigma, s' + 2\ell', 0) + \ell'(2S - 3)$.*

Proof: by induction on s' and ℓ' .

If $\ell' = 0$, then there is only one thrifty packing of σ . By Proposition 12, $R(\sigma, s', 0) = OPT(\sigma, s', 0)$.

If $s' = 0$, then any reasonable packing puts one item of size L into each bin that it uses. By Proposition 10, $OPT(\sigma, 2\ell', 0) \geq 2\ell'S$. Since each bin in σ has size at most $2L - 1$, it follows that $R(\sigma, 0, \ell') \leq \ell'(2L - 1) = \ell'(4S - 3) \leq OPT(\sigma, 2\ell', 0) + \ell'(2S - 3)$.

Let $s', \ell' \geq 1$ and suppose the claim is true for s'' and ℓ'' , if $0 \leq s'' < s'$ or $0 \leq \ell'' < \ell'$. Let σ be any sequence of bins, let $B \leq M = 4S - 3$ be the size of the first bin in σ , and let σ' be obtained from σ by removing its first bin. Since $s', \ell' \geq 1$, it follows that $s' + 2\ell' \geq 3 \geq \lfloor B/S \rfloor$. Note that $OPT(\sigma, s' + 2\ell', 0) = B + OPT(\sigma', s' + 2\ell' - \lfloor B/S \rfloor, 0)$, because OPT packs $\lfloor B/S \rfloor$ items of size S in the first bin.

Consider any thrifty algorithm. If it packs the first bin with only items of size S , then, it packs $\lfloor B/S \rfloor$ items into that bin and the total space it uses is at most $B + R(\sigma', s' - \lfloor B/S \rfloor, \ell')$, which, by the induction hypothesis, is at most $B + OPT(\sigma', s' + 2\ell' - \lfloor B/S \rfloor, 0) + \ell'(2S - 3) = OPT(\sigma, s' + 2\ell', 0) + \ell'(2S - 3)$. So assume that, in the first bin, the algorithm packs one item of size L plus possibly one item of size S .

If $B < L + S$, then the algorithm packs no items of size S into the first bin and uses at most $B + R(\sigma', s', \ell' - 1)$ space. By the induction hypothesis, $R(\sigma', s', \ell' - 1) \leq OPT(\sigma', s' + 2\ell' - 2, 0) + (\ell' - 1)(2S - 3)$. Since OPT packs at most two items of size S into the first bin, $B + OPT(\sigma', s' + 2\ell' - 2, 0) \leq OPT(\sigma, s' + 2\ell', 0)$. Hence, the space used by the algorithm is at most $B + R(\sigma', s', \ell' - 1) \leq OPT(\sigma, s' + 2\ell', 0) + \ell'(2S - 3)$.

Otherwise, $L + S \leq B \leq M = 2L - 1$. Since the algorithm is thrifty, it also packs one item of size S into the first bin. Then the space used by the algorithm is at most $B + R(\sigma', s' - 1, \ell' - 1) \leq B + OPT(\sigma', s' - 1 + 2\ell' - 2, 0) + (\ell' - 1)(2S - 3)$, by the induction hypothesis. Since OPT packs at most three items of size S into the first bin, $B + OPT(\sigma', s' + 2\ell' - 3, 0) \leq OPT(\sigma, s' + 2\ell', 0)$. Hence, the space used by the algorithm is at most $B + R(\sigma', s' - 1, \ell' - 1) \leq OPT(\sigma, s' + 2\ell', 0) + \ell'(2S - 3)$.

It follows that, in all cases, $R(\sigma, s', \ell') \leq OPT(\sigma, s' + 2\ell', 0) + \ell'(2S - 3)$. ■

Lemma 14 *For all sequences of bins σ and all integers $s', \ell' \geq 0$, if $2k = s' + 2\ell'$, then $R(\sigma, s', \ell') \leq OPT(\sigma, 0, k) + (s' + \ell' - k - 1)L + M$.*

Proof: If $s' = 0$, the lemma follows from Proposition 12. So, we assume that $s' \geq 1$. Let σ be a sequence of bins that maximizes $R(\sigma, s', \ell') - OPT(\sigma, 0, k)$. Suppose that, among all such sequences, σ has the smallest capacity (i.e. the smallest sum of bin sizes).

Consider a reasonable packing with cost $R(\sigma, s', \ell')$ on σ . If σ has any bins of size in $[S, L - 1]$, they are not used by OPT and, by the minimality of the capacity of σ , they are used by the reasonable packing to pack one item of size S . All other bins have size at least L and are used by both the reasonable packing and OPT, unless one of them has already packed all its items.

Consider the last bin in σ . It must be used by either OPT or the reasonable packing; otherwise, removing it would decrease the capacity of the sequence without changing the total space used by either packing, contradicting the definition of σ . If it is only used by the reasonable packing, it must have size M ; otherwise replacing it by a bin of size M would increase the total space used by the reasonable packing without changing the total space used by OPT, contradicting the definition of σ .

If the last bin is only used by OPT (to pack one item of size L), it must have size L ; otherwise replacing it by a bin of size L would decrease the total space used by OPT without changing the total space used by the reasonable packing, contradicting the definition of σ .

If the last bin is used by OPT and the reasonable packing to each pack one item, it must have size L ; otherwise replacing it by a bin of size L would decrease the capacity of the sequence while changing the total space used by OPT and the reasonable packing by the same amount, contradicting the definition of σ .

Finally, the last bin cannot be used by the reasonable packing to pack more than one item; otherwise it must have size between $2S$ and M . If it is replaced by a bin of size L followed by a bin of size M , the total space used by the reasonable packing would increase, and the space used by OPT would not. This contradicts the definition of σ .

Now suppose that σ has a bin, other than its last bin, that does not have size L . Let b be the last such bin. Let s'' and ℓ'' be the number of items of size S and L , respectively, that the reasonable packing has remaining immediately after packing bin b . These items are packed one per bin in each of the next $s'' + \ell''$ bins. Since the packing is reasonable, the items of size L are packed before the items of size S . If k'' is the number of items (of size L) that OPT has remaining immediately after bin b , they are packed one per bin in each of the next k'' bins.

If $\text{size}(b) \leq L - 1$, then the reasonable packing packs a single item of size S in this bin. Increasing the size of b to L does not change the bins that the reasonable packing uses: it still packs one item in each of the $1 + s'' + \ell''$ bins starting with b , since all of them, except possibly the last, have size L . This increases the total space used by the reasonable packing. However, the total space used by OPT remains unchanged. This is because if OPT has any items remaining immediately before bin b is packed, they are each packed in a bin of size L . This contradicts the definition of σ . Since $\text{size}(b) \neq L$, it follows that $\text{size}(b) > L$.

If bin b is only used by OPT (to pack one item of size L), then replacing it by a bin of size L would decrease the total space used by OPT without changing the total space used by the reasonable packing, contradicting the definition of σ . Therefore the reasonable packing packs at least one item in bin b .

Suppose that OPT uses bin b . If the reasonable packing packs one item in bin b , then decreasing the size of bin b to L would decrease the capacity of the sequence while changing the total cost incurred by OPT and the reasonable packing by the same amount, contradicting the definition of σ . Hence the reasonable packing packs $h > 1$ items in bin b . Decreasing the size of bin b to L and adding $h - 1$ bins of size L immediately following bin b decreases the total space used by OPT by $\text{size}(b) - L > 0$, since OPT still packs each item following bin b in a bin of size L . It also increases the total space used by the reasonable packing by $hL - \text{size}(b) > 0$, since it packs one item in each bin, from bin b onwards. This contradicts the definition of σ . Therefore, OPT does not use bin b . Since $\text{size}(b) > L$, OPT runs out of items before bin b .

Consider the last bin b' in σ prior to which OPT has at least one item (of size L) available and the reasonable packing has items of both size S and size L available. Then, up to including when bin b' is packed, twice the number of items of size L that OPT has available is at least the number of items of size S plus twice the number of items of size L that the reasonable packing has available. In particular, it is true before the first bin of σ is packed, since $2k = s' + \ell'$. For each bin of size at least L , up to and including bin b' , OPT packs one item of size L and the reasonable algorithm packs at least one item of size L or at least two items of size S , so the inequality remains true. Moreover, OPT packs no items in bins whose size is less than L , so such bins do not cause the inequality to become false.

It follows that if OPT runs out of items immediately after bin b' , then the reasonable packing does too, which contradicts the fact that the reasonable packing uses bin b , but OPT has run out of items prior to bin b . If the reasonable packing runs out of items of size S immediately after bin b' , then OPT has at least as many items of size L left as the reasonable packing. Thus, it does not run out of items before the reasonable packing. This is also a contradiction.

Therefore, immediately before bin b' and, hence, immediately before bin b , the reasonable packing only has items of size S left. Since bin b is not the last bin, the reasonable packing is thrifty, and $\text{size}(b) > L$, the reasonable packing has at least two items in b . Replace bin b by a bin of size $b - S \geq S$ followed by a bin of size L . Then the reasonable packing must pack one fewer item of size S in the first of these bins and one item of size S in the second. This adds $L - S$ to the total space used by the reasonable packing, but does not change the total space used by OPT, contradicting the definition of σ .

Thus, we may assume that σ consists of $s' + \ell' - 1 \geq k$ bins of size L , followed by at most one bin of size M . Then $\text{OPT}(\sigma, 0, k) = kL$ and $R(\sigma, s', \ell') \leq (s' + \ell' - 1)L + M$, so $R(\sigma, s', \ell') - \text{OPT}(\sigma, 0, k) \leq (s' + \ell' - k - 1)L + M$. ■

Lemma 15 *For all sequences of bins σ and all integers $s', \ell' \geq 0$, if $k \geq s' + 2\ell'$, then $R(\sigma, s' + 2\ell', 0) \leq \text{OPT}(\sigma, s', \ell') + (k - s' - \ell' - 1)L + M$.*

Proof: by induction on s' and ℓ' .

If $\ell' = 0$, then the first s' items of a reasonable packing are packed the same as OPT would pack them. Since any reasonable packing is thrifty, each bin that contains one of the remaining $k - s'$ items of size S has waste at most $S - 1$, except for the last such bin, which has waste at most $M - S$. Thus $R(\sigma, k, 0) \leq \text{OPT}(\sigma, s', 0) + (k - s')S + (k - s' - 1)(S - 1) + M - S = \text{OPT}(\sigma, s', 0) + (k - s' - 1)L + M$.

Next consider $s' = 0$. Any thrifty packing of k items of size S has waste at most $S - 1$ in each bin that it uses, except the last, which has waste at most $M - S$. Since it uses at most k bins, its cost is at most $kS + (k - 1)(S - 1) + M - S = (k - 1)L + M$. By Proposition 10, $\text{OPT}(\sigma, 0, \ell') \geq \ell'L$, so $R(\sigma, k, 0) \leq (k - 1)L + M = \ell'L + (k - \ell' - 1)L + M \leq \text{OPT}(\sigma, 0, \ell') + (k - \ell' - 1)L + M$.

Now, let $s', \ell' > 0$ and suppose the claim is true if $s'' < s'$ or $\ell'' < \ell'$. Let σ be any sequence of bins, and let $B \leq M = 4S - 3$ be the size of its first bin. Since $k \geq s' + 2\ell' \geq 3$, any reasonable packing of k items of size S puts $\lfloor B/S \rfloor \leq 3$ items into this bin, so has cost at most $B + R(\sigma', k - \lfloor B/S \rfloor, 0)$, where σ' is the sequence σ without its first bin.

If OPT packs the first bin with only items of size S , then, it packs $\lfloor B/S \rfloor \leq s'$ items into that bin and the total space it uses is $B + \text{OPT}(\sigma', s' - \lfloor B/S \rfloor, \ell')$. By the induction hypothesis, $B + R(\sigma', k - \lfloor B/S \rfloor, 0) \leq B + \text{OPT}(\sigma', s' - \lfloor B/S \rfloor, \ell') + (k - \lfloor B/S \rfloor - (s' - \lfloor B/S \rfloor) - \ell' - 1)L + M = \text{OPT}(\sigma, s', \ell') + (k - s' - \ell' - 1)L + M$.

Otherwise OPT packs the first bin with one item of size L and $\lfloor (B - L)/S \rfloor \geq \lfloor B/S \rfloor - 2$ items of size S . Since $k - \lfloor B/S \rfloor \geq s' + 2\ell' - \lfloor (B - L)/S \rfloor - 2 = (s' - \lfloor (B - L)/S \rfloor) + 2(\ell' - 1)$, it follows by the induction hypothesis that $B + R(\sigma', k - \lfloor B/S \rfloor, 0) \leq B + \text{OPT}(\sigma', s' - \lfloor (B - L)/S \rfloor, \ell' - 1) + (k - \lfloor B/S \rfloor - (s' - \lfloor B/S \rfloor - 2)) - (\ell' - 1) - 1)L + M \leq \text{OPT}(\sigma, s', \ell') + (k - s' - \ell' - 1)L + M$.

Hence, $R(\sigma, k, 0) \leq B + R(\sigma', k - \lfloor B/S \rfloor, 0) \leq \text{OPT}(\sigma, s', \ell') + (k - s' - \ell' - 1)L + M$. ■

Lemma 16 *For all sequences of bins σ and all integers $s', \ell' \geq 0$, if $2k \geq s' + 2\ell'$, then $R(\sigma, 0, k) \leq \text{OPT}(\sigma, s', \ell') + \min\{0, \ell' - s'\}S + (k - \ell')M$.*

Proof: by induction on s' and ℓ' .

If $s' = 0$, then the first ℓ' items are packed the same way in a reasonable packing as OPT would pack them. Each of the remaining $k - \ell'$ items of size L are packed one per bin using bins of size at most M , for a total cost at most $\text{OPT}(\sigma, 0, \ell') + (k - \ell')M = \text{OPT}(\sigma, 0, \ell') + \min\{0, \ell' - s'\}S + (k - \ell')M$, since $\ell' - s' \geq 0$.

If $\ell' = 0$, then, by Proposition 10, $\text{OPT}(\sigma, s', \ell') \geq s'S + \ell'L = \max\{0, s' - \ell'\}S$, so $\text{OPT}(\sigma, s', \ell') + \min\{0, \ell' - s'\}S \geq 0$. In a reasonable packing, each of the k items of size L are packed one per bin using bins of size at most M for a total cost at most $kM \leq \text{OPT}(\sigma, s', \ell') + \min\{0, \ell' - s'\}S + (k - \ell')M$.

Now, let $s', \ell' \geq 1$ and suppose the claim is true if $s'' < s'$ or $\ell'' < \ell'$. Let σ be any sequence of bins, let $B \leq M = 4S - 3$ be the size of its first bin, and let σ' denote the same sequence as σ , but with the first bin removed.

If $B < L$, then a reasonable packing leaves the first bin of σ empty, since it has no items of size S . Hence, it uses space at most $R(\sigma', 0, k)$. Since $2k \geq s' + 2\ell' \geq (s' - 1) + 2\ell'$, the induction hypothesis implies that $R(\sigma', 0, k) \leq \text{OPT}(\sigma', s' - 1, \ell') + \min\{0, \ell' - (s' - 1)\}S + (k - \ell')M$. OPT packs one item of size S into the first bin of σ , so $\text{OPT}(\sigma, s', \ell') = B + \text{OPT}(\sigma', s' - 1, \ell')$. Thus, since $S \leq B$, it follows that $R(\sigma', 0, k) \leq \text{OPT}(\sigma', s' - 1, \ell') + B - S + \min\{0, \ell' - (s' - 1)\}S + (k - \ell')M \leq \text{OPT}(\sigma, s', \ell') + \min\{0, \ell' - s'\}S + (k - \ell')M$.

Otherwise, $B \geq L$. Then, after packing the first bin of σ , a reasonable packing has $k - 1$ items (of size L) remaining. Suppose OPT has s'' items of size S and ℓ'' items of size L remaining after packing this bin. Hence, by the induction hypothesis, to pack its $k - 1$ items of size L , any reasonable packing uses space at most $R(\sigma', 0, k - 1) \leq \text{OPT}(\sigma', s'', \ell'') + \min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M$, since $2(k - 1) \geq s' + 2(\ell' - 1) = (s' - 2) + 2\ell' \geq s'' + 2\ell''$. Therefore, $R(\sigma, 0, k) = B + R(\sigma', 0, k - 1) \leq B + \text{OPT}(\sigma', s'', \ell'') + \min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M = \text{OPT}(\sigma, s', \ell') + \min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M$.

Since OPT is reasonable and $B \leq M = 2L - 1 = 4S - 3$, either $\ell'' = \ell' - 1$ and $s'' \geq s' - 1$ or $\ell'' = \ell'$ and $s'' \geq s' - 3$. In the first case, $\ell'' - s'' \leq (\ell' - 1) - (s' - 1) = \ell' - s'$ and $k - 1 - \ell'' = k - \ell'$, so $\min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M \leq \min\{0, \ell' - s'\}S + (k - \ell')M$. In the second case, $\ell'' - s'' \leq \ell' - s' + 3$ and $k - 1 - \ell'' = k - \ell' - 1$, so $\min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M \leq \min\{0, \ell' - s'\}S + 3S + (k - \ell')M - M \leq \min\{0, \ell' - s'\}S + (k - \ell')M$, since $3S \leq 4S - 3 = M$, provided $S \geq 3$. If $S = 2$, then $M = 5$, so $s'' \geq s' - 2$ and $2S < M$, so $\min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M \leq \min\{0, \ell' - s'\}S + 2S + (k - \ell')M - M \leq \min\{0, \ell' - s'\}S + (k - \ell')M$. Thus, in all cases, $R(\sigma, 0, k) \leq \text{OPT}(\sigma, s', \ell') + \min\{0, \ell'' - s''\}S + (k - 1 - \ell'')M \leq \text{OPT}(\sigma, s', \ell') + \min\{0, \ell' - s'\}S + (k - \ell')M$. ■

5 Simple Non-Optimal Algorithms

It is helpful to understand why simple reasonable algorithms are not optimal for the Restricted Grid Scheduling problem. The following examples show why a number of natural candidates do not work well enough.

Example 1. Consider the reasonable algorithm that always uses items of size S , when there is a choice. Let $s = 2\ell$ and let σ consist of ℓ bins of size $2S$, followed by s bins of size S , and then ℓ bins of size M . For this instance, the algorithm has a performance ratio of $\frac{\ell \cdot 2S + \ell \cdot M}{\ell \cdot 2S + s \cdot S} = \frac{3}{2} - \frac{3}{4S}$, which is greater than $5/4$ for large S .

Example 2. Consider the reasonable algorithm that always uses items of size L , when there is a choice. Let $s = 2\ell$ and let σ consist of ℓ bins of size $2S$, followed by $s - 1$ bins of size L and then 1 bin of size M . For this instance, the algorithm has a performance ratio of $\frac{\ell \cdot 2S + (s-1) \cdot L + M}{\ell \cdot 2S + \ell \cdot L} = \frac{3}{2} + \frac{1}{2\ell} - \frac{3/\ell + 1}{2(4S-1)}$, which is also greater than $5/4$ for large S .

For the two instances considered above, the reasonable algorithm which alternates between using two items of size S and one item of size L , when it has a choice, would do well, achieving a performance ratio of $5/4$. However, it does not do as well on other instances.

Example 3. Let $2s = 3\ell$ and let σ consist of ℓ bins of size $2S$, followed by s bins of size S and $\ell/2$ bins of size M . For this instance, the algorithm which alternates between using two items of size S and one item of size L , when it has a choice, has a performance ratio of $\frac{\ell \cdot 2S + (s-\ell) \cdot S + \ell/2 \cdot M}{\ell \cdot 2S + s \cdot S} = 1 + \frac{\frac{\ell}{2} \cdot (M-2S)}{(2\ell+s) \cdot S} = 1 + \frac{2}{7} - \frac{3}{7S}$, which is larger than $5/4$ for S sufficiently large.

As the above examples partially illustrate, once either the online algorithm or OPT has run out of one type of item (items of size S or items of size L), the adversary can give bins which make the online algorithm waste a lot of space. The algorithm we present in the next section aims to postpone this situation long enough to get a good ratio.

When $3s = 2\ell$, the lower bound is in Case II. It starts by giving the algorithm $\lfloor s/2 \rfloor$ bins of size $2S$ and it considers the number k of these bins in which the algorithm packs two items of size S . The best the algorithm can do is to have $k \approx \max\{\lfloor s/2 \rfloor - s/8 - \ell/4 + 1, \lfloor s/2 \rfloor - s/3 + 1\} = \lfloor s/2 \rfloor - s/3 + 1 \approx s/6$, so that the same ratio is obtained for both subcases. This is the same as using one item of size L twice as often as two items of size S , when it has a choice.

Example 4. Let $3s = 2\ell$. Consider the reasonable algorithm which uses one item of size L twice as often as two items of size S , when it has a choice. Let σ consist of ℓ bins of size $2S$, followed by s bins of size S and $\ell/3$ bins of size M . For this instance, the algorithm packs $\ell/3 = s/2$ bins of size $2S$ with two items of size S , so it has a performance ratio of $\frac{\ell \cdot 2S + (\ell/3) \cdot M}{\ell \cdot 2S + s \cdot S} = 1 + \frac{\frac{\ell}{3} \cdot M - sS}{(2\ell+s) \cdot S} = 1 + \frac{S-3/2}{4 \cdot S} = \frac{5}{4} - \frac{3}{8S}$, which exceeds the lower bound of $1 + \frac{2}{3+6\ell/s} = \frac{7}{6}$ for S sufficiently large.

This example shows that it is not always possible to achieve the lower bound by choosing to use two items of size S instead of one item of size L a fixed proportion of the time. After $s/6$ bins have been packed with two items of size S , it turns out to be better to always use an item of size L , when there is a choice. This motivates our use of a second phase to obtain the best competitive ratio. The number of items of size S that the algorithm has used, in this case $s/6 \times 2 = s/3$, indicates when the first phase is done.

6 A Matching Upper Bound

In this section, we present a reasonable algorithm, 2-Phase-Packer, for the Restricted Grid Scheduling problem. It is asymptotically optimal: the competitive ratio matches the lower bound in Section 3 for all three ranges of the ratio s/ℓ of the initial numbers of items of size S and items of size L . Not surprisingly, 2-Phase-Packer has two phases.

In the first phase, it attempts to balance the number of items of size S and the number of items of size L it uses, aiming for the ratio indicated by the lower bound. When it receives bins where it has a choice of using one item of size L or two items of size S in part or all of that bin (i.e., bins with sizes in the ranges $[2S, 3S - 2]$ and $[3S, 4S - 3]$), it uses one item of size L in a certain fraction of them (and increments the variable L-bins) and uses two items of size S in the remaining fraction (and increments S-bins). The fraction varies depending on the original ratio s/ℓ : at least 2, between 2 and $6/5$, and less than $6/5$. It is enforced by a macro called UseLs, which indicates that an item of size L should be used if and only if $\text{L-bins} \leq r \text{S-bins}$, where r is the target ratio of L-bins to S-bins. For example, when equal numbers of each should be used, we have $\text{UseLs} = (\text{L-bins} \leq \text{S-bins})$. Both L-bins and S-bins start at zero, so, in this case, 2-Phase-Packer starts by choosing to use one item of size L and then alternates. Note that S-bins and L-bins do not change after Phase 1 is completed.

In the middle range for the ratio s/ℓ , there are two different fractions used. The first fraction is used until S-bins reaches a specific value. Afterwards, its choices alternate. To do so, for the rest of Phase 1, it records the number of times it chooses to pack two items of size S in a bin and the number of times it chooses to pack one item of size L in late-S-bins and late-L-bins, respectively. The variables late-S-bins and late-L-bins are also zero initially.

2-Phase-Packer uses countS and countL throughout the algorithm to keep track of the total numbers of items of size S and items of size L it has used, whether or not it had a choice. (Specifically, countS is incremented every time an item of size S is used and countL is incremented every time an item of size L is used.) It continues with Phase 1 until it has used a certain number of items of size S or items of size L (depending on the relationship between s and ℓ). For each of the three ranges of s/ℓ , we define a different condition for ending Phase 1. In Phase 2, only items of size S or only items of size L are used where a reasonable algorithm has a choice, depending on whether one would expect an excess of items of size S or items of size L , given the ratio s/ℓ .

The definition of the algorithm implies the following relationships.

Observation 17 $L\text{-bins} + \text{late-}L\text{-bins} \leq \text{countL}$ and $2(S\text{-bins} + \text{late-}S\text{-bins}) \leq \text{countS}$.
If $\ell \leq s/2$ or $5s/6 < \ell$, then $\text{late-}S\text{-bins} = \text{late-}L\text{-bins} = 0$.

The definitions of the end of Phase 1 for the various ranges of s/ℓ imply these inequalities.

Lemma 18 If $s/2 < \ell \leq 5s/6$, then $S\text{-bins} + \text{late-}S\text{-bins} \leq 3s/8 - \ell/4 + 1$.
If $5s/6 < \ell$, then $S\text{-bins} \leq s/6 + 1$.

The following simple invariants can be proved inductively.

```

macro Phase1done =  $\begin{cases} \text{countL} \geq \lfloor \ell/2 \rfloor & \text{if } \ell \leq s/2 \\ \text{countS} \geq \lfloor 3s/4 - \ell/2 \rfloor & \text{if } s/2 < \ell \leq 5s/6 \\ \text{countS} \geq \lfloor s/3 \rfloor & \text{if } 5s/6 < \ell \end{cases}$ 

macro UseLs =  $\begin{cases} \text{L-bins} \leq \text{S-bins} & \text{if } \ell \leq s/2 \\ \text{if } (\text{S-bins} < \lfloor (s + 2\ell)/16 \rfloor) & \text{if } s/2 < \ell \leq 5s/6 \\ \quad \text{then L-bins} \leq (10\ell - 3s)\text{S-bins}/(s + 2\ell) \\ \quad \text{else late-L-bins} \leq \text{late-S-bins} \\ \text{L-bins} \leq 2\text{S-bins} & \text{if } 5s/6 < \ell \end{cases}$ 

countS % counts the number of items of size  $S$  used; initially 0
countL % counts the number of items of size  $L$  used; initially 0
S-bins  $\leftarrow$  L-bins  $\leftarrow$  late-S-bins  $\leftarrow$  late-L-bins  $\leftarrow$  0

for each arriving bin  $b$ 
  if only one type of item still remains then use as many items as fit in  $b$ 
  else if  $\text{size}(b) \in [S, 2S - 2]$  then use 1 item of size  $S$ 
  else if  $\text{size}(b) = 2S - 1$  then use 1 item of size  $L$ 
  else if  $\text{size}(b) = 3S - 1$  then use 1 item of size  $L$  and 1 item of size  $S$ 
  else if only one item of size  $S$  still remains
    then use 1 item of size  $L$ 
    if remaining space in  $b$  is at least  $S$  then use 1 item of size  $S$ 
  else if only two items of size  $S$  still remain and  $\text{size}(b) \in [3S, 4S - 3]$ 
    then use 1 item of size  $L$  and 1 item of size  $S$ 
  else if (not Phase1done)
    then % Use range determined ratio
    if  $\text{size}(b) \in [2S, 3S - 2]$  then
      if UseLs then use 1 item of size  $L$ 
      else use 2 items of size  $S$ 
    else %  $\text{size}(b) \in [3S, 4S - 3]$ 
      if UseLs then use 1 item of size  $L$  and 1 item of size  $S$ 
      else use 3 items of size  $S$ 
    if  $s/2 < \ell \leq 5s/6$  and  $\text{S-bins} \geq \lfloor (s + 2\ell)/16 \rfloor$  then
      if UseLs then late-L-bins ++
      else late-S-bins ++
    else %  $\ell \leq s/2$  or  $\ell > 5s/6$  or  $\text{S-bins} < \lfloor (s + 2\ell)/16 \rfloor$ 
      if UseLs then L-bins ++
      else S-bins ++
  else % In Phase 2
    if  $\ell \leq s/2$  then use as many items of size  $S$  as fit in bin  $b$ 
    else use 1 item of size  $L$ 
    if remaining space in  $b$  is at least  $S$  then use 1 item of size  $S$ 
end for

```

Figure 2: The algorithm 2-Phase-Packer

Lemma 19 *If $\ell \leq s/2$, then $S\text{-bins} \leq L\text{-bins} \leq S\text{-bins} + 1$.
If $s/2 < \ell \leq 5s/6$, then $\text{late-}S\text{-bins} \leq \text{late-}L\text{-bins} \leq \text{late-}S\text{-bins} + 1$ and
 $\lfloor c(S\text{-bins} - 1) \rfloor + 1 \leq L\text{-bins} \leq \lfloor c S\text{-bins} \rfloor + 1$, where $1 < c = \frac{10\ell - 3s}{s + 2\ell} \leq 2$.
If $5s/6 < \ell$, then $2S\text{-bins} \leq L\text{-bins} \leq 2S\text{-bins} + 1$.*

We analyse the performance of 2-Phase-Packer as compared to the cost of an optimal reasonable packing on an arbitrary sequence, σ , using the three different ranges for the relationship between s and ℓ in the lower bound.

Both 2-Phase-Packer and OPT use exactly the same bins until one of them runs out of either items of size L or items of size S . Thus, there are four cases to consider.

1. OPT runs out of items of size L at or before the point where 2-Phase-Packer runs out of anything.
2. OPT runs out of items of size S at or before the point where 2-Phase-Packer runs out of anything.
3. 2-Phase-Packer runs out of items of size L before OPT runs out of anything.
4. 2-Phase-Packer runs out of items of size S before OPT runs out of anything.

We examine each of these four cases and, within each case, we consider three different ranges for the relationship between s and ℓ .

Consider an arbitrary sequence of bins σ in which 2-Phase-Packer packs s items of size S and ℓ items of size L . Consider the first bin b' after which either OPT or 2-Phase-Packer had only one type of item remaining. Up to and including bin b' , both OPT and 2-Phase-Packer have unpacked items of both sizes. Let $\text{count}L'$ and $\text{count}S'$ denote the number of items of size L and items of size S , respectively, that 2-Phase-Packer has used up to and including bin b' . Let $S\text{-bins}'$ and $\text{late-}S\text{-bins}'$ be the values of $S\text{-bins}$ and $\text{late-}S\text{-bins}$ immediately after bin b' , so $S\text{-bins}' + \text{late-}S\text{-bins}'$ denotes the number of bins at or before b' where 2-Phase-Packer had a choice and used two items of size S instead of one of size L . Similarly, $L\text{-bins}' + \text{late-}L\text{-bins}'$ denotes the number in which 2-Phase-Packer used one item of size L instead of two items of size S , where $L\text{-bins}'$ and $\text{late-}L\text{-bins}'$ are the values of $L\text{-bins}$ and $\text{late-}L\text{-bins}$ immediately after bin b' . Let σ' denote the portion of σ after bin b' .

Both algorithms use all bins up to and including bin b' , since all bins have size at least S . Let X be the total cost of these bins. Then $X \geq (\text{count}S')S + (\text{count}L')L$ and $2\text{-Phase-Packer}(\sigma, s, \ell) = X + 2\text{-Phase-Packer}(\sigma', s - \text{count}S', \ell - \text{count}L')$.

Case 1: OPT has no items of size L after b' . In this case, Lemma 9 implies that, after bin b' , OPT has $(s - \text{count}S') + 2(\ell - \text{count}L')$ items of size S remaining, so $OPT(\sigma, s, \ell) = X + OPT(\sigma', (s - \text{count}S') + 2(\ell - \text{count}L'), 0)$.

Since 2-Phase-Packer is reasonable, Lemma 13 implies that $2\text{-Phase-Packer}(\sigma', s - \text{count}S', \ell - \text{count}L') \leq OPT(\sigma', (s - \text{count}S') + 2(\ell - \text{count}L'), 0) + (\ell - \text{count}L')(2S - 3)$, so

$$2\text{-Phase-Packer}(\sigma, s, \ell) \leq OPT(\sigma, s, \ell) + (\ell - \text{count}L')(2S - 3).$$

By Proposition 10, $\text{OPT}(\sigma, s, \ell) \geq sS + \ell L \geq sL/2 + \ell L = (s + 2\ell)L/2$. When computing the competitive ratio, we will choose the additive constant to be at least L , so we subtract L from 2-Phase-Packer's cost in the ratio. Thus,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - L}{\text{OPT}(\sigma, s, \ell)} &\leq \frac{\text{OPT}(\sigma, s, \ell) + (\ell - \text{countL}') (2S - 3) - L}{\text{OPT}(\sigma, s, \ell)} \\ &\leq 1 + \frac{(\ell - \text{countL}' - 1)(L - 2)}{(s + 2\ell)L/2} \\ &\leq 1 + \frac{2(\ell - \text{countL}' - 1)}{s + 2\ell}. \end{aligned}$$

It remains to bound $\ell - \text{countL}' - 1$ in each of the three ranges for the ratio s/ℓ . We use the fact that, immediately after bin b' , OPT has run out of items of size L , so at least ℓ bins of size at least L are in σ up to and including bin b' .

First, consider the case when $\ell \leq s/2$. If $\text{countL}' < \lfloor \ell/2 \rfloor$, then Phase 1 was not completed when bin b' arrived. Therefore, each time a bin of size at least L arrives up to and including bin b' , 2-Phase-Packer either packs an item of size L in it and, hence, increments countL , or it increments S-bins. Hence, $\text{countL}' + \text{S-bins}' \geq \ell$. By Lemma 19, $\text{L-bins}' \geq \text{S-bins}'$. Since $\text{countL}' \geq \text{L-bins}'$, by Observation 17, it follows that $\text{countL}' \geq \ell/2 \geq \lfloor \ell/2 \rfloor$. This is a contradiction. Thus, $\text{countL}' \geq \lfloor \ell/2 \rfloor$ and

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - L}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2(\ell - \text{countL}' - 1)}{s + 2\ell} \\ &\leq 1 + \frac{\ell}{s + 2\ell} = 1 + \frac{1}{2 + s/\ell}. \end{aligned}$$

So, suppose that $s/2 < \ell$. During Phase 1, each time a bin of size at least L arrives, 2-Phase-Packer either packs an item of size L in it and, hence, increments countL , or it increments one of S-bins or late-S-bins. During Phase 2, 2-Phase-Packer packs an item of size L in each such bin until it runs out of items of size L . Therefore, $\text{countL}' + \text{S-bins}' + \text{late-S-bins}' \geq \ell$.

If $\ell \leq 5s/6$, then Lemma 18 implies that $\ell - \text{countL}' \leq \text{S-bins}' + \text{late-S-bins}' \leq 3s/8 - \ell/4 + 1$ and

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - L}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2(\ell - \text{countL}' - 1)}{s + 2\ell} \leq 1 + \frac{2(3s/8 - \ell/4)}{s + 2\ell} \\ &= 1 + \frac{s/4 + \ell/2 + s/2 - \ell}{s + 2\ell} < 1 + \frac{s/4 + \ell/2}{s + 2\ell} = \frac{5}{4}. \end{aligned}$$

Similarly, if $s < 6\ell/5$, then $\text{late-S-bins}' = 0$, by Observation 17, so $\ell - \text{countL}' \leq \text{S-bins}' \leq s/6 + 1$, by Lemma 18, and

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - L}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2(\ell - \text{countL}' - 1)}{s + 2\ell} \\ &\leq 1 + \frac{s/3}{s + 2\ell} < 1 + \frac{2}{3 + 6\ell/s}. \end{aligned}$$

Case 2: OPT has no items of size S after b' . In this case, Lemma 9 implies that, after bin b' , OPT has $k = (s - \text{countS}')/2 + (\ell - \text{countL}')$ items of size L remaining, so $\text{OPT}(\sigma, s, \ell) = X + \text{OPT}(\sigma', 0, k)$.

Since 2-Phase-Packer is reasonable, Lemma 14 implies that

$$\begin{aligned} 2\text{-Phase-Packer}(\sigma, s, \ell) &= X + 2\text{-Phase-Packer}(\sigma', s - \text{countS}', \ell - \text{countL}') \\ &\leq X + \text{OPT}(\sigma', 0, k) + ((s - \text{countS}') + (\ell - \text{countL}') - k - 1)L + M \\ &= \text{OPT}(\sigma, s, \ell) + (s - \text{countS}')L/2 + M - L. \end{aligned}$$

By Proposition 10, $\text{OPT}(\sigma, s, \ell) \geq sS + \ell L \geq sL/2 + \ell L$.

When computing the competitive ratio, we will choose the additive constant to be at least M , so we subtract M from 2-Phase-Packer's cost in the ratio. Thus,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq \frac{\text{OPT}(\sigma, s, \ell) + (s - \text{countS}')L/2 - L}{\text{OPT}(\sigma, s, \ell)} \\ &\leq 1 + \frac{((s - \text{countS}')/2 - 1)L}{sL/2 + \ell L} \\ &= 1 + \frac{s - \text{countS}' - 2}{s + 2\ell}. \end{aligned}$$

It remains to bound $s - \text{countS}' - 2$ in each of the three ranges for the ratio s/ℓ .

First, suppose that $\ell \leq s/2$. Immediately after bin b' , OPT has packed at most $2L\text{-bins}'$ additional items of size S than 2-Phase-Packer, so $s \leq \text{countS}' + 2L\text{-bins}'$. When Phase 1 is finished, $\text{countL} = \lfloor \ell/2 \rfloor$, since countL either changes by 0 or 1 each iteration. After Phase 1, $L\text{-bins}$ does not change. By Observation 17, $L\text{-bins} \leq \text{countL}$. Hence $L\text{-bins}' \leq \lfloor \ell/2 \rfloor \leq \ell/2$. Thus,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{s - \text{countS}' - 2}{s + 2\ell} < 1 + \frac{2L\text{-bins}'}{s + 2\ell} \\ &\leq 1 + \frac{\ell}{s + 2\ell} = 1 + \frac{1}{2 + s/\ell}. \end{aligned}$$

Next, consider the case where $s/2 < \ell \leq 5s/6$. If $\text{countS}' \geq 3s/4 - \ell/2 - 2$, then $s - \text{countS}' - 2 \leq s - 3s/4 + \ell/2 = (s + 2\ell)/4$. Otherwise, $\text{countS}' \leq 3s/4 - \ell/2 - 2 < \lfloor 3s/4 - \ell/2 \rfloor$, so Phase 1 is not completed when bin b' is packed. Therefore, the number of bins in which 2-Phase-Packer packed one item of size L , but OPT packed two items of size S is at most $L\text{-bins}' + \text{late-L-bins}'$. Since OPT has used all of the items of size S , $s \leq \text{countS}' + 2L\text{-bins}' + 2\text{late-L-bins}'$. By Observation 17 and Lemma 19, $2S\text{-bins}' + 2\text{late-S-bins}' \leq \text{countS}'$, $L\text{-bins}' \leq \lfloor (10\ell - 3s)S\text{-bins}'/(s + 2\ell) \rfloor + 1$, and $\text{late-L-bins}' \leq \text{late-S-bins}' + 1$. From the algorithm, $S\text{-bins}' \leq \lfloor (s + 2\ell)/16 \rfloor$, since $S\text{-bins}$ increases by at most one each iteration until it reaches $\lfloor (s + 2\ell)/16 \rfloor$ and does not increase thereafter. Hence $s - \text{countS}' - 2 \leq 2(L\text{-bins}' + \text{late-L-bins}' - 1) \leq 2(1 + (10\ell - 3s)S\text{-bins}'/(s + 2\ell) + 1 + \text{late-S-bins}' - 1) = 2(S\text{-bins}' + \text{late-S-bins}' + (8\ell - 4s)S\text{-bins}'/(s + 2\ell) + 1) \leq \text{countS}' + 2(8\ell - 4s)/16 + 2 \leq 3s/4 - \ell/2 + \ell - s/2 = s/4 + \ell/2 = (s + 2\ell)/4$. Therefore,

$$\frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} \leq 1 + \frac{s - \text{countS}' - 2}{s + 2\ell} \leq 1 + \frac{(s + 2\ell)/4}{s + 2\ell} = \frac{5}{4}.$$

When $5s/6 < \ell$, we show that $\text{countS}' \geq \lfloor s/3 \rfloor$. Suppose not. Then Phase 1 is not completed when bin b' is packed, so the number of bins in which 2-Phase-Packer packed one item of size L , but OPT packed two items of size S is at most $\text{L-bins}'$. Since OPT has used all of the items of size S , $s \leq \text{countS}' + 2\text{L-bins}'$. By Lemma 19 and Observation 17, $\text{L-bins}' \leq 2\text{S-bins}' + 1 \leq \text{countS}' + 1$. Hence $s \leq 3\text{countS}' + 2$ and $\text{countS}' \geq (s - 2)/3$. Since countS' is an integer and $(s - 2)/3 > (s - 3)/3 \geq \lfloor s/3 \rfloor - 1$ it follows that $\text{countS}' \geq \lfloor s/3 \rfloor$, which is a contradiction.

Thus, $\text{countS}' \geq \lfloor s/3 \rfloor$, so $s - \text{countS}' - 2 \leq \lceil 2s/3 \rceil - 2 < 2s/3$ and

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{s - \text{countS}' - 2}{s + 2\ell} \\ &< 1 + \frac{2s/3}{s + 2\ell} \\ &= 1 + \frac{2}{3 + 6\ell/s}. \end{aligned}$$

Case 3: 2-Phase-Packer runs out of items of size L before OPT runs out of anything.

Suppose that, in this instance, 2-Phase-Packer packs its last item of size L in bin b' , but, after bin b' , OPT has $i > 0$ items of size S and $j > 0$ items of size L that are unpacked. Then, by Lemma 9, 2-Phase-Packer has $s - \text{countS}' = i + 2j$ unpacked items of size S .

Note that $j \leq \text{L-bins}' + \text{late-L-bins}'$, since the only bins where 2-Phase-Packer had a choice and used one item of size L where OPT possibly did not are the bins contributing to the values of L-bins and late-L-bins . By Lemma 15, $2\text{-Phase-Packer}(\sigma, s, \ell) = X + 2\text{-Phase-Packer}(\sigma', i + 2j, 0) \leq X + \text{OPT}(\sigma', i, j) + (j - 1)L + M = \text{OPT}(\sigma, s, \ell) + (j - 1)L + M$. By Proposition 10, $\text{OPT}(\sigma, s, \ell) \geq sS + \ell L \geq sL/2 + \ell L$. In computing the competitive ratio, we will let the additive constant be M , so we will subtract this value from 2-Phase-Packer's cost in the ratio.

Thus,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq \frac{\text{OPT}(\sigma, s, \ell) + (j - 1)L}{\text{OPT}(\sigma, s, \ell)} \\ &= 1 + \frac{(j - 1)L}{\text{OPT}(\sigma, s, \ell)} \\ &\leq 1 + \frac{(j - 1)L}{sL/2 + \ell L} \\ &= 1 + \frac{2j - 2}{s + 2\ell} \end{aligned}$$

We now bound $2j$ in each of the three cases of the algorithm.

Suppose $\ell \leq s/2$. Phase 1 has completed, since all items of size L have been used by 2-Phase-Packer. $\text{L-bins}'$ is at most the value of countL when Phase 1 completed, which is $\lfloor \ell/2 \rfloor$. Thus, $2j \leq$

$2L\text{-bins}' \leq 2\lfloor \ell/2 \rfloor \leq \ell$. Hence,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2j}{s + 2\ell} \\ &\leq 1 + \frac{\ell}{s + 2\ell} \\ &\leq 1 + \frac{1}{2 + s/\ell}. \end{aligned}$$

Next, suppose that $6\ell/5 \leq s < 2\ell$. If $\text{countS}' \geq 3s/4 - \ell/2 - 2$, then $2j - 2 = s - \text{countS}' - i - 2 < s/4 + \ell/2 = (s + 2\ell)/4$. Otherwise, $\text{countS}' < 3s/4 - \ell/2 - 2 < \lfloor 3s/4 - \ell/2 \rfloor$, so Phase 1 is not done when bin b' is packed. Therefore, j , the number of bins in which 2-Phase-Packer packed one item of size L , but OPT did not, is at most $L\text{-bins}' + \text{late-L-bins}'$. By Observation 17 and Lemma 19, $2S\text{-bins}' + 2\text{late-S-bins}' \leq \text{countS}'$, $L\text{-bins}' \leq \lfloor (10\ell - 3s)S\text{-bins}'/(s + 2\ell) \rfloor + 1$, and $\text{late-L-bins}' \leq \text{late-S-bins}' + 1$. From the algorithm, $S\text{-bins}' \leq \lfloor (s + 2\ell)/16 \rfloor$, since $S\text{-bins}'$ increases by at most one each iteration until it reaches $\lfloor (s + 2\ell)/16 \rfloor$ and does not increase thereafter. Hence $2j - 2 \leq 2(L\text{-bins}' + \text{late-L-bins}' - 1) \leq 2(1 + (10\ell - 3s)S\text{-bins}'/(s + 2\ell) + 1 + \text{late-S-bins}' - 1) = 2(S\text{-bins}' + \text{late-S-bins}' + (8\ell - 4s)S\text{-bins}'/(s + 2\ell) + 1) \leq \text{countS}' + 2(8\ell - 4s)/16 + 2 < 3s/4 - \ell/2 - 2 + \ell - s/2 + 2 = (s + 2\ell)/4$. Hence,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2j - 2}{s + 2\ell} \\ &\leq 1 + \frac{(s + 2\ell)/4}{s + 2\ell} = \frac{5}{4}. \end{aligned}$$

Now, suppose that $6\ell/5 > s$. If Phase 1 is done, then $\text{countS}' \geq \lfloor s/3 \rfloor > s/3 - 1$, so $2j = s - \text{countS}' - i \leq s - s/3 + 1 - i \leq 2s/3$. Otherwise, $\text{countS}' < \lfloor s/3 \rfloor$. Then, since $\text{countS}' \geq 2S\text{-bins}'$, it follows from Lemma 19 that $j \leq L\text{-bins}' \leq 2S\text{-bins}' + 1 \leq \text{countS}' + 1 \leq \lfloor s/3 \rfloor \leq s/3$. Hence,

$$\begin{aligned} \frac{2\text{-Phase-Packer}(\sigma, s, \ell) - M}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{2j}{s + 2\ell} \\ &\leq 1 + \frac{2s/3}{s + 2\ell} \\ &< 1 + \frac{2}{3 + 6\ell/s}. \end{aligned}$$

Case 4: 2-Phase-Packer runs out of items of size S before OPT runs out of anything.

Suppose that, in this instance, 2-Phase-Packer packs its last item of size S in bin b' , but, after bin b' , OPT has $i > 0$ items of size S and $j > 0$ items of size L that are unpacked. Then, 2-Phase-Packer has $\ell - \text{countL}'$ unpacked items of size L , and, by Lemma 9, $2(\ell - \text{countL}') = i + 2j$. Note that, in this case, $\ell \geq i/2 + j \geq \frac{1}{2} + 1$, so $\ell \geq 2$.

Since 2-Phase-Packer is reasonable, Lemma 16 implies that

$$\begin{aligned} 2\text{-Phase-Packer}(\sigma, s, \ell) &= X + R(\sigma', 0, \ell - \text{countL}') \\ &\leq X + \text{OPT}(\sigma', i, j) + \min\{0, j - i\}S + (\ell - \text{countL}' - j)M \\ &= \text{OPT}(\sigma, s, \ell) + \min\{0, j - i\}S + (i/2)M. \end{aligned}$$

By Proposition 10, $\text{OPT}(\sigma, s, \ell) \geq sS + \ell L \geq sL/2 + \ell L$. Thus,

$$\begin{aligned}
\frac{\text{2-Phase-Packer}(\sigma, s, \ell)}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{(i/2)M + \min\{0, j - i\}S}{sL/2 + \ell L} \\
&= 1 + \frac{i(2L - 1) + \min\{0, j - i\}(L + 1)}{sL + 2\ell L} \\
&= 1 + \frac{L(i + \min\{i, j\}) - i + \min\{0, j - i\}}{L(s + 2\ell)} \\
&\leq 1 + \frac{i + \min\{i, j\}}{s + 2\ell}.
\end{aligned}$$

First suppose $\ell \leq s/2$. If Phase 1 has not ended, then the only bins where 2-Phase-Packer had a choice and used two more items of size S than OPT did are the bins contributing to the value of S-bins, so $i \leq 2\text{S-bins}'$. By Lemma 19, $\text{S-bins}' \leq \text{L-bins}' \leq \text{countL}'$.

If $j \geq i$, then $2(\ell - \text{countL}') = i + 2j \geq 3i$, so $i \leq 2(\ell - \text{countL}')/3$. In this case, if $\text{countL}' \geq \ell/4$, then $i \leq 2(\ell - \text{countL}')/3 \leq \ell/2$. If not, since $\ell \geq 2$, $\text{countL}' < \ell/4 \leq (\ell - 1)/2 \leq \lfloor \ell/2 \rfloor$, so Phase 1 has not ended and $i \leq 2\text{S-bins}' \leq 2\text{L-bins}' \leq 2\text{countL}' < \ell/2$. Therefore,

$$\frac{\text{2-Phase-Packer}(\sigma, s, \ell)}{\text{OPT}(\sigma, s, \ell)} \leq 1 + \frac{2i}{s + 2\ell} \leq 1 + \frac{\ell}{s + 2\ell} = 1 + \frac{1}{2 + s/\ell}.$$

Otherwise, $j < i$. If $\text{countL}' \geq \lfloor \ell/2 \rfloor$, then $i + j = 2(\ell - \text{countL}') - j \leq 2(\ell - \text{countL}') - 1 \leq 2\lfloor \ell/2 \rfloor - 1 \leq \ell$. If not, $\text{countL}' < \lfloor \ell/2 \rfloor$, so Phase 1 has not ended, $i \leq 2\text{S-bins}' \leq 2\text{L-bins}' \leq 2\text{countL}'$, and $i + j = \ell - \text{countL}' + i/2 \leq \ell$. Hence

$$\frac{\text{2-Phase-Packer}(\sigma, s, \ell)}{\text{OPT}(\sigma, s, \ell)} \leq 1 + \frac{i + j}{s + 2\ell} \leq 1 + \frac{\ell}{s + 2\ell} \leq 1 + \frac{1}{2 + s/\ell}.$$

So, suppose that $\ell > s/2$. Then $i \leq 2(\text{S-bins}' + \text{late-S-bins}')$, because the only bins where 2-Phase-Packer had a choice and used two more items of size S than OPT did are the bins contributing to the value of S-bins or late-S-bins. Since 2-Phase-Packer has run out of items of size S , Phase 1 has ended.

Now suppose that $s/2 < \ell \leq 5s/6$. If $\text{S-bins}' + \text{late-S-bins}' \leq (s + 2\ell)/16$, then $i + \min\{i, j\} \leq 2i \leq 4(\text{S-bins}' + \text{late-S-bins}') \leq (s + 2\ell)/4$, so

$$\frac{\text{2-Phase-Packer}(\sigma, s, \ell)}{\text{OPT}(\sigma, s, \ell)} \leq 1 + \frac{s + 2\ell}{4(s + 2\ell)} = \frac{5}{4}.$$

Otherwise, $\text{S-bins}' = \lfloor (s + 2\ell)/16 \rfloor$. By Lemma 19, $\text{L-bins}' \geq \lfloor c(\text{S-bins}' - 1) \rfloor + 1$, where $1 < c =$

$\frac{10\ell-3s}{s+2\ell} \leq 2$ and $\text{late-L-bins}' \geq \text{late-S-bins}'$. Since $\text{L-bins}' + \text{late-L-bins}' \leq \text{countL}'$, it follows that

$$\begin{aligned}
i + \min\{i, j\} \leq i + j &= \frac{i}{2} + \ell - \text{countL}' \\
&\leq \text{S-bins}' + \text{late-S-bins}' + \ell - \text{L-bins}' - \text{late-L-bins}' \\
&\leq \ell + \text{S-bins}' - \lfloor c(\text{S-bins}' - 1) \rfloor - 1 \\
&< \ell + \text{S-bins}' - c(\text{S-bins}' - 1) \\
&= \ell + (1 - c)\lfloor (s + 2\ell)/16 \rfloor + c \\
&\leq \ell + (1 - c)((s + 2\ell)/16 - 1) + c \\
&= \ell + \left(1 - \frac{10\ell - 3s}{s + 2\ell}\right) \frac{s + 2\ell}{16} + 2c - 1 \\
&= (4s + 8\ell)/16 + 2c - 1 \text{ and} \\
\frac{2\text{-Phase-Packer}(\sigma, s, \ell) - 3}{\text{OPT}(\sigma, s, \ell)} &\leq 1 + \frac{(4s + 8\ell)/16}{s + 2\ell} = \frac{5}{4}.
\end{aligned}$$

Finally, suppose that $5s/6 < \ell$. When Phase 1 finished, $2\text{S-bins} \leq \text{countS} \leq \lfloor s/3 \rfloor + 2$. Thereafter, S-bins does not change. Hence, $i \leq 2\text{S-bins}' \leq s/3 + 2$ and

$$\frac{2\text{-Phase-Packer}(\sigma, s, \ell) - 4}{\text{OPT}(\sigma, s, \ell)} \leq 1 + \frac{2i - 4}{s + 2\ell} \leq 1 + \frac{2s/3}{s + 2\ell} = 1 + \frac{2}{3 + 6\ell/s}.$$

7 Conclusions and Open Problems

We have shown that varying the proportion of items of size S to items of size L does not lead to a larger competitive ratio if the maximum bin size is at most $4S - 3$. This may also be the case for an arbitrary maximum bin size, but there are complications. First, in bins of size $2kS$, where $k \geq 2S - 1$, it is possible to pack more than k items of size L . In addition, it may be an advantage to have mixed bins which contain more than one item of size S and more than one item of size L . So, when bins can be large, one needs to consider how to maintain a good ratio of items of size L to items of size S , as they are used. We conjecture that maintaining the ratios specified in 2-Phase-Packer is sufficient.

One could also consider the Grid Scheduling problem for two bin sizes, S and $L \neq 2S - 1$. For example, when L is a multiple of S , then the algorithm of Example 2, which always uses as many items of size L as possible, is optimal and, hence, has competitive ratio 1. It would be interesting to see if changing the ratio of S to L could improve the lower bound. We conjecture that it does not.

For the general problem, where there could be more than two sizes of items, we would like to close the gap between our lower bound and the upper bound of $\frac{13}{7}$ in [1]. Using many item sizes, it is not hard to prove a lower bound of $\frac{3}{2}$ on the strict competitive ratio (the competitive ratio where the additive constant in the definition is zero). However, the strict competitive ratio is not very interesting for this problem, since the ratio can be made larger simply by increasing the size of M and giving a last bin of size M .

Finally, it would be interesting to consider the competitive ratio of randomized algorithms for the Grid Scheduling problem or Restricted Grid Scheduling problem against an oblivious adversary.

References

- [1] J. Boyar and L. M. Favrholdt. Scheduling jobs on Grid processors. *Algorithmica*, 57(4):819–847, 2010.
- [2] J. Boyar and L. M. Favrholdt. A new variable-sized bin packing problem. *Journal of Scheduling*, 15:273–287, 2012.
- [3] A. Darling, L. Carey, and W. Feng. The design, implementation, and evaluation of mpiBLAST. In *ClusterWorld Conference & Expo and the 4th International Conference on Linux Cluster: The HPC Revolution 2003*, 2003.
- [4] L. Epstein, L. M. Favrholdt, and A. Levin. Online variable-sized bin packing with conflicts. *Discrete Optimization*, 8(2):333–343, 2011.
- [5] L. Epstein and A. Levin. More online bin packing with two item sizes. *Discrete Optimization*, 5(4):705–713, 2008.
- [6] G. Gutin, T. R. Jensen, and A. Yeo. On-line bin packing with two item sizes. *Algorithmic Operations Research*, 1(2), 2006.
- [7] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. *Algorithmica*, 3(1):79–119, 1988.
- [8] F. M. Liang. A lower bound for on-line bin packing. *Inform. Process. Lett.*, 10:76–79, 1980.
- [9] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Comm. of the ACM*, 28(2):202–208, 1985.
- [10] G. Zhang. A new version of on-line variable-sized bin packing. *Discrete Applied Mathematics*, 72:193–197, 1997.